STATUS OF THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Original) A method to provide a platform-level network security framework comprising:

identifying a packet associated with a processor system;

identifying one or more platform-level network security protocols associated with an extensible firmware interface (EFI); and

identifying the packet with a network security condition based on the one or more platform-level network security protocols.

2. (Original) A method as defined in claim 1, wherein identifying the packet associated with the processor system comprises identifying at least one of an incoming packet and outgoing packet during at least one of a pre-boot environment and a post-boot environment.

3. (Original) A method as defined in claim 1, wherein identifying the packet associated with the processor system comprises identifying an incoming packet from a network interface and an outgoing packet from an operating system.

4. (Original) A method as defined in claim 1, wherein identifying the one or more platform-level network security protocols associated with the EFI comprises identifying the one or more protocols of at least one configuration table associated with at least one of a firewall, a virtual private network, and an Internet Protocol Security framework.

5. (Original) A method as defined in claim 1, wherein identifying the one or more platform-level network security protocols associated with the EFI comprises identifying the one or more platform-level network security protocols associated with the EFI based on a configuration table having one or more global unique identifiers and one or more data pointers.

6. (Original) A method as defined in claim 1, wherein identifying the packet with a network security condition based on the one or more platform-level network security protocols comprises associating the packet with at least one of an allowable condition and a deny condition based on one or more protocols of at least one configuration table associated with at least one of a firewall, a virtual private network, and an Internet Protocol Security framework.

7. (Original) A method as defined in claim 1 further comprising transmitting the packet to a protocol stack in response to identifying the packet with an allowable condition based on the one or more platform-level network security protocols.

8. (Original) A method as defined in claim 1 further comprising discarding the packet in response to identifying the packet with a deny condition based on the one or more platform-level network security protocols.

9. (Original) A machine readable medium storing instructions which, when executed, cause a machine to:

identify a packet associated with a processor system;

identify one or more platform-level network security protocols associated with an extensible firmware interface (EFI); and

identify the packet with a network security condition based on the one or more platform-level network security protocols.

10. (Original) A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to identify the packet associated with the processor system by identifying at least one of an incoming packet and outgoing packet during at least one of a pre-boot environment and a post-boot environment.

11. (Original) A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to identify the packet associated with the processor system by identifying at least one of an incoming packet from a network interface and an outgoing packet from an operating system.

12. (Original) A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to identify the one or more platform-level network security protocols associated with the EFI by identifying the one or more protocols of a configuration table associated with at least one of a firewall, a virtual private network, and an Internet Protocol Security framework.

13. (Original) A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to identify the one or more platform-level network security protocols associated with the EFI by identifying the one or more platform-level network security protocols associated with the EFI based on a configuration table having one or more global unique identifiers and one or more data pointers.

14. (Original) A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to identify the packet with a network security condition based on the one or more platform-level network security protocols by identifying the packet with at least one of an allowable condition and a deny condition based on one or more protocols of at least one configuration table associated with at least one of a firewall, a virtual private network, and an Internet Protocol Security framework.

15. (Original) A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to transmit the packet to a protocol stack in response to identifying the packet with an allowable condition based on the platform-level network security protocols.

16. (Original) A machine readable medium as defined in claim 9, wherein the instructions, when executed, cause the machine to discard the packet in response to identifying the packet with a deny condition based on the one or more platform-level network security protocols.

17. (Original) An apparatus to provide a platform-level network security framework comprising:

a network interface to communicate packets;

an interrupt handler coupled to the network interface to receive an interrupt request (IRQ); and

a network interface driver coupled to the interrupt handler to identify a packet associated with a processor system, to identify one or more platform-level network security protocols associated with an extensible firmware interface (EFI), and to identify the packet with a network security condition based on the one or more platform-level network security protocols.

18. (Original) An apparatus as defined in claim 17, wherein the network interface comprises a network interface card.

19. (Original) An apparatus as defined in claim 17, wherein the network interface driver is to identify at least one of an incoming packet from the network interface and an outgoing packet from an operating system during at least one of a pre-boot environment and a post-boot environment.

20. (Original) An apparatus as defined in claim 17, wherein the network interface driver is to transmit the packet to a protocol stack in response to identifying the packet with an allowable condition based on the platform-level network security protocols.

21. (Original) An apparatus as defined in claim 17, wherein the network interface driver discards the packet in response to identifying the packet with a deny condition based on the one or more platform-level network security protocols.

22. (Original) An apparatus as defined in claim 17, wherein the one or more platform-level network security protocols comprises one or more protocols of at least one configuration table associated with at least one of a firewall, a virtual private network, and an Internet Protocol Security framework.

23. (Original) An apparatus as defined in claim 17 configuration table having one or more globally unique identifiers and one or more data pointers to identify the one or more the platform-level network security protocols.

24. (Original) A processor system to provide a platform-level network security framework comprising:

a network interface to communicate packets; and

a processor coupled to the network interface, the processor programmed to identify a packet associated with the processor system, to identify one or more platform-level network security protocols associated with an extensible firmware interface (EFI), and to identify the packet with a network security condition based on the one or more platform-level network security protocols.

25. (Original) A processor system as defined in claim 24, wherein the network interface comprises a network interface card.

26. (Original) A processor system as defined in claim 24, wherein the processor is programmed to identify at least one of an incoming packet from the network interface and an outgoing packet from an operating system during at least one of a pre-boot environment and a post-boot environment.

27. (Original) A processor system as defined in claim 24, wherein the processor is programmed to transmit the packet to a protocol stack in response to identifying packet with an allowable condition based on the platform-level network security protocols.

28. (Original) A processor system as defined in claim 24, wherein the processor is programmed to discard the packet in response to identifying the packet with a deny condition based on the one or more platform-level network security protocols.

29. (Original) A processor system as defined in claim 24, wherein the one or more platform-level network security protocols comprises one or more protocols of at least one configuration table associated with at least one of a firewall, a virtual private network, and an Internet Protocol Security framework.

30. (Original) A processor system as defined in claim 24 further comprising a configuration table having one or more globally unique identifiers and one or more data pointers to identify the one or more the platform-level network security protocols.